

УДК 519.873:519.718.7

Л.А. Золоторевич

МОДЕЛИРОВАНИЕ НЕИСПРАВНОСТЕЙ В СТРУКТУРАХ СБИС НА ЯЗЫКЕ VHDL

Рассматривается проблема моделирования неисправностей СБИС и устройств цифровой электроники, представленных на структурном уровне на языке VHDL. Дается анализ эффективности предлагаемых решений.

Введение

По мере развития микроэлектроники актуальность и сложность решения задачи разработки тестов для верификации проектов и контроля изделий на всех этапах жизненного цикла постоянно возрастают. В настоящее время разработчики СБИС желают строить тесты на различных уровнях описания проектов на языке VHDL, в том числе и на уровне поведенческого описания. Ряд исследователей предлагают рассматривать поведенческие неисправности объектов как программные ошибки и соответственно применять методы мутирования исходных кодов, разработанные для тестирования программного обеспечения [1], однако при этом не обосновывается соответствие этих кодов реальным физическим неисправным модификациям объекта проектирования.

В настоящей работе обобщаются результаты моделирования неисправностей применительно к схемному представлению устройства, а также рассматривается задача моделирования неисправностей константного типа и определения полноты теста для объекта, описанного на языке VHDL. Архитектурное тело проекта представляет устройство на структурном уровне.

1. Моделирование неисправностей при структурном представлении устройства

Известно, что на начальном этапе развития средств автоматизации в электронике решение задачи контроля сводилось к проверке правильности реализуемых устройством функций и построению функциональных тестов. Особенность данного подхода заключается в сложности построения функционального теста, обеспечивающего требуемую достоверность контроля объекта на этапах проектирования, изготовления и эксплуатации, а также в необходимости проведения анализа его полноты в заданном классе неисправностей, характерном для применяемой технологии. К настоящему времени найдено удовлетворительное решение задачи построения тестов контроля неисправностей, описывающих определенные физические дефекты реальных объектов. Особенно широко исследован класс константных неисправностей. К сожалению, подход, основанный на структурном представлении объекта, не позволяет находить эффективные практические решения задачи генерации тестов последовательностных устройств практической размерности. Подобные решения были найдены для объектов, разработанных с учетом требований контролепригодности. При этом применялись в основном методы, направленные на обработку комбинационных структур.

Как правило, практические подходы к генерации тестов из-за сложности методов направленного поиска основаны на моделировании объектов диагностирования. При этом большинство разработанных методов ориентировано на рассмотрение структурного представления цифрового устройства [2–4]. Применительно к структурным представлениям цифровых устройств разработаны методы и инструментальные средства моделирования с внесением (инъекцией) одиночных неисправностей, параллельного моделирования неисправностей, когда за один проход моделируется исправная схема и «пачка» неисправностей (размер «пачки» определяется разрядной сеткой моделирующей ЭВМ), дедуктивного (сквозного) моделирования неисправностей. Наиболее эффективным из известных и практически реализованных методов является метод сквозного вычисления множеств контролируемых неисправностей, основанный на моделировании только исправной схемы и не требующий собственно моделирования ее неисправных модификаций [4]. Данный метод основан на применении теоретико-множественных соотношений (табл. 1), позволяющих вычислять множества

неисправностей константного типа, обнаруживаемых на выходах вентильных элементов и функционально-сложных блоков, представленных на автоматном уровне таблицами переходов-выходов.

На основе приведенных соотношений разработаны программные средства, достаточно эффективно решающие задачи анализа полноты тестов и генерации тестов цифровых структур методом интерпретативного моделирования псевдослучайных последовательностей [5]. Наиболее эффективным является применение разработанных средств при условии использования знаний о функциональном назначении внешних выводов исследуемого устройства, а также и без привлечения эксперта при генерации тестов для контролепригодных объектов.

Таблица 1

Базовые соотношения для разработки системы анализа тестов на полноту контроля

Функция, реализуемая элементом	Теоретико-множественные соотношения для вычисления множеств обнаруживаемых неисправностей
AND	$F_{(f=0)} = \left(\bigcap_{i \in M_0} A_i \setminus \bigcup_{j \in M_1} A_j \right) \cup e_1; \quad F_{(f=1)} = \bigcup_{i \in M_1} A_i \cup e_0,$ <p>где M_0 – множество входов с логическим нулем в исправном состоянии; M_1 – множество входов с логической единицей в исправном состоянии; A_i, A_j – множества неисправностей, обнаруживаемых на i-м и j-м входах элемента; e_1, e_0 – внутренние неисправности элемента константного типа</p>
NAND	$F_{(f=1)} = \left(\bigcap_{i \in M_0} A_i \setminus \bigcup_{j \in M_1} A_j \right) \cup e_0; \quad F_{(f=0)} = \bigcup_{i \in M_1} A_i \cup e_1$
OR	$F_{(f=1)} = \left(\bigcap_{i \in M_1} A_i \setminus \bigcup_{j \in M_0} A_j \right) \cup e_0; \quad F_{(f=0)} = \bigcup_{i \in M_0} A_i \cup e_1$
NOR	$F_{(f=0)} = \left(\bigcap_{i \in M_1} A_i \setminus \bigcup_{j \in M_0} A_j \right) \cup e_0; \quad F_{(f=1)} = \bigcup_{i \in M_0} A_i \cup e_0$
Произвольная функция, заданная таблицей переходов-выходов	$F = \bigcup_{f=1, S} \left(\bigcup_{p=1, M} \left(\bigcap_{i \in Q_1} A_i \setminus \bigcup_{j \in Q_0} A_j \right) \cup e^* \right),$ <p>где Q_1 – множество входных переменных и переменных внутреннего состояния, значения которых инверсны состоянию в исправной схеме на данном входном наборе; Q_0 – множество входных переменных и переменных внутреннего состояния, значения которых совпадают со значениями переменных в исправной схеме; e^* – внутренняя неисправность блока, обнаруживаемая на выходе рассматриваемого элемента на данном входном наборе; M – количество строк в таблице, соответствующих выходным векторам, отличным от выходного вектора при отсутствии неисправностей; S – длина анализируемого теста</p>

2. Структурное моделирование неисправностей СБИС на основе языка VHDL

В последнем десятилетии определился новый подход к проектированию в микроэлектронике, основанный на применении языка описания и моделирования высокого уровня VHDL. В рамках данного подхода процесс нисходящего проектирования осуществляется автоматически (автоматизированно), начиная с формулировки задания на проектирование в общем виде и создания проекта на системном уровне и кончая его детализацией на этапе топологического проектиро-

вания. При таком подходе проект изделия на функционально-логическом и последующих уровнях является результатом автоматизированного синтеза и представляет собой автоматически полученное описание устройства на языке VHDL. В этом случае исходными данными для моделирования является программа функционирования проектируемой схемы, написанная на языке VHDL.

Переход на использование языка VHDL требует переосмысления и решения задач анализа контролепригодности проектируемого объекта, контроля тестов на полноту и построения проверяющих тестов, так как известные методы решения данных задач в условиях применения VHDL оказались непригодными для применения на практике.

Постоянное повышение степени интеграции, структурной сложности СБИС, переход на проектирование и изготовление СБИС по принципу «система на чипе» требуют решения указанных выше задач на разных уровнях проектирования, при этом желательно их решение постепенно перенести на самый высокий уровень с целью сокращения сроков проектирования изделия в целом.

Данная работа направлена на моделирование неисправностей СБИС на основе языка VHDL, когда архитектурное тело проекта описывает структуру объекта (объект представлен как взаимосвязь компонент).

Известны два подхода к моделированию неисправностей на структурном уровне описания объектов на языке VHDL [6–8]. Первый основан на введении *диверсантов* в описание компоненты и замене исходных постоянных компонент так называемыми мутантами. Предполагается изменение (мутация) описания архитектурного тела, которое будет происходить автоматически или вручную с целью отображения функционирования устройства при появлении определенной неисправности. При этом полученные «мутанты» ведут себя идентично оригинальным компонентам, за исключением интервала времени при введении неисправности. Данный подход применяется в работе [6] и позволяет описывать неисправности широкого класса. При этом задача моделирования неисправностей решается на уровне выбора определенных конфигураций базовых компонент.

Второй подход к введению неисправностей состоит в манипулировании переменными и сигналами модели во время процесса моделирования с использованием встроенных команд системы моделирования. В работе [6] описан инструментальный комплекс MERISTO, который базируется на обоих указанных выше подходах.

В работе [7] предлагается программная система VERIFY, которая позволяет вводить неисправности в цифровую систему на различных уровнях абстракции. Она основывается на динамических мутантах, где описание неисправности является составной частью поведенческого описания компонент. Для этих целей язык VHDL был расширен с тем, чтобы можно было описать тип, частоту появления и среднее время существования неисправности. Для того чтобы можно было вводить неисправности в соответствии с этим описанием, разработан компилятор и система моделирования. Естественный путь обмена информацией с компонентой в VHDL – это применение сигналов. Поэтому была использована концепция сигнала для описания неисправности, связанной с компонентой и в то же время с имитатором, чтобы активизировать неисправность в заданное время. Каждая из возможных неисправностей соответствующей компоненты может быть описана отдельным сигналом. Имеется две возможности внесения неисправностей. Первая из них – сделать сигналы введения неисправностей FIS (fault injection signals) и их параметры видимыми для системы моделирования, включив сигналы в описание интерфейса (entity), вторая – сохранить FIS прозрачными для других компонент. В первом случае FIS должны быть описаны как порты VHDL в интерфейсе компоненты. Это требовало бы сделать FIS всех поведенческих компонент цифровой схемы видимыми на верхнем уровне иерархии системы. Для каждого из FIS должен быть задан «путь» через все уровни иерархии. В работе обеспечена полная прозрачность описания неисправностей каждой компоненты. С этой целью введен новый для синтаксиса VHDL сигнал. В данном случае FIS описываются как внутренние сигналы, имеющие расширение на два дополнительных параметра – среднее время встречаемости неисправности и ее среднюю длительность, к примеру:

SIGNAL i_stuck_at_0: BOOLEAN INTERVAL 10000 h DURATION 5 ns.

После того как исходный код модели системы скомпилирован, выполнение всех экспериментов по введению неисправностей осуществляется автоматически без взаимодействия с пользователем системы. Система моделирования независимо вводит требуемое количество неисправ-

ностей. Время и местоположение следующей неисправности определяются системой моделирования автоматически. Этот подход гарантирует, что если неисправность возникает в два раза чаще, чем другая, то и вводится она будет в два раза чаще. Следовательно, контролепригодность системы может быть оценена введением нескольких тысяч неисправностей во время моделирования.

Рассмотрим задачу анализа полноты теста и моделирования неисправностей, если архитектурное тело устройства содержит структурное описание объекта (рис. 1).

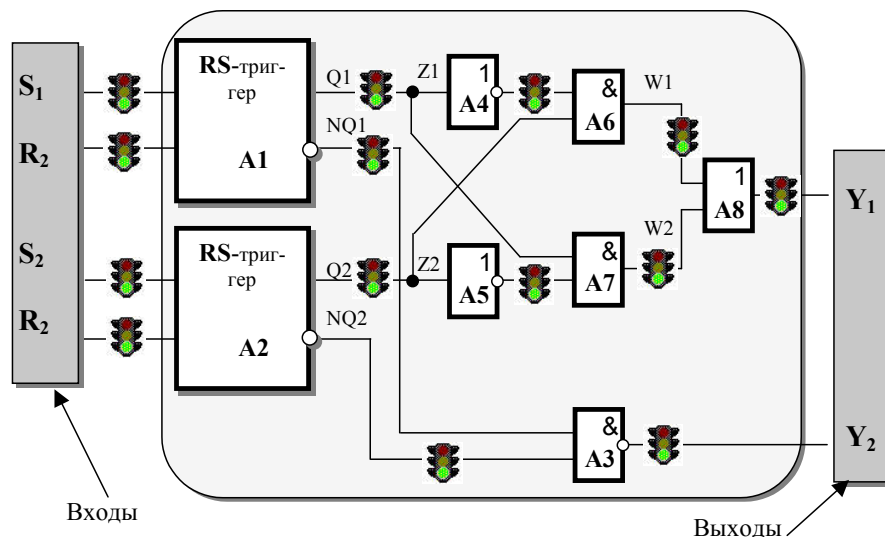


Рис. 1. Пример цифрового устройства

Общая схема анализа полноты теста приведена на рис. 2. Поставим задачу инъекции и моделирования константных неисправностей без необходимости многократной перекомпиляции описания. Для этого в описание исправного объекта на языке VHDL внедрим блоки-имплантаты (рис. 1), которые наделим простой функцией с внешним управлением искажения определенного сигнала в заданное время моделирования. Возможность внешнего управления активизацией имплантатов на этапе моделирования позволяет разрабатывать модели неисправностей, не требующие перекомпиляции проекта для инъекции каждой неисправности. На рис. 3 показаны результаты моделирования неисправностей схемы (рис. 1) в системе ModelSim фирмы Mentor Graphics. Внешнее управление процессом инъекции неисправностей осуществляется добавлением в описание устройства дополнительных переменных (рис. 4).

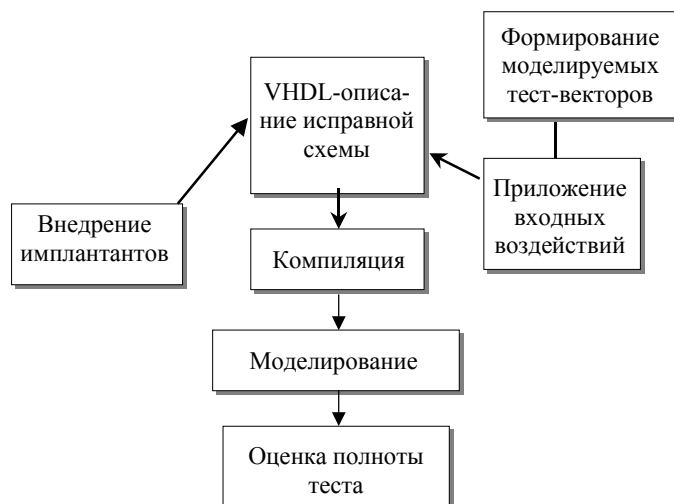


Рис. 2. Общая структура процесса моделирования неисправностей

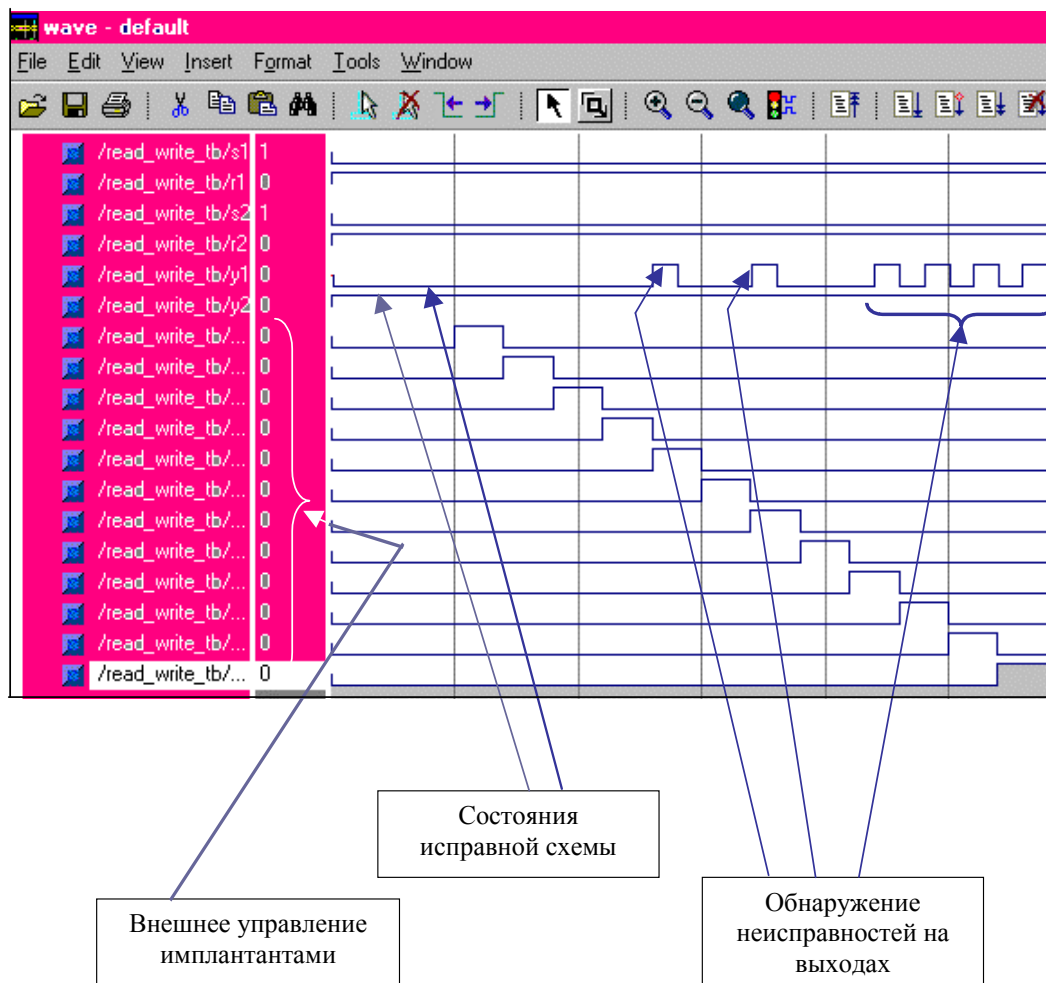



Рис. 3. Результаты моделирования



```

entity error is
    port(in_port,c:in std_logic;
    out_port:inout std_logic);
end error;

ARCHITECTURE error_arch OF error IS
    shared variable tm: integer;
    BEGIN
        T: process (in_port,c)
        BEGIN
            if c='0'
                then out_port<=in_port;
                else out_port<='0','1' after tm*1 ns;
            end if;
        end process T;
    end error_arch;
  
```

Рис. 4. Блок имплантанта на языке VHDL

Заключение

Приведены основные теоретико-множественные соотношения для наиболее эффективного решения задачи моделирования неисправностей СБИС на уровне функционально-логического представления. Дается анализ известных подходов к моделированию неисправностей структур СБИС на основе VHDL. Рассматривается методика моделирования неисправностей константного типа на основе VHDL-описаний исправного устройства без необходимости многократной компиляции описания.

Список литературы

1. Hayek Al., Robach G. On the adequacy of deriving hardware test data from the behavioral specification // Proc. EUROMICRO 96, 22nd euromicro conference. – 1996. – P. 337–342.
2. Zolotorevitch L.A., Baturitsky M.A. Deductive switch-level CMOS-VLSI fault simulation // The International conference computer-aided design of discrete devices (CAD DD'95). V. 2. – Minsk-Szczecin, 1995. – P. 157–164.
3. Armstrong D. A deductive method for simulating faults in logic circuits // IEEE trans. comput. – 1972. – C-21, 5.
4. Золоторевич Л.А., Сидоренко О.М., Юхневич Д.И. Формализация и оптимизация анализа полноты теста СБИС на уровне автоматных моделей компонентов // Мат. Междунар. конф. «Автоматизация проектирования дискретных систем (CAD-DD'97)». Т. 3. – Мн.: Ин-т техн. кибернетики НАН Беларуси, 1997. – С. 104–110.
5. Золоторевич Л.А., Сидоренко О.М., Юхневич Д.И. Компьютерная инженерия в электронике: программная система для проектирования и обучения // Мат. Междунар. науч.-практ. конф. «Образовательные технологии в подготовке специалистов». – Мн., 2003. – С. 23–28.
6. Jenn E., Arlat J., Rimen M. Fault injection into VHDL models: The MEFISTO tool // Proc. 24th symp. on fault tolerant comp. (FTCS-24). – Austin, Texas, USA, 1994. – P. 66–75.
7. Sieh V., Tschache O., Balbach F. VERIFY: evaluation of reliability using VHDL-models with embedded fault descriptions // Proc. 27th Int. symp. on fault-tolerant comp. (FTCS-27). – Chicago, 1997. – P. 32–36.
8. Goswami K.K., Iyer R.K. A simulation-based study of a triple modular redundant system using DEPEND // Proc. of the 5th Int. conference on fault-tolerant computing systems. – Paris, 1991. – P. 300–311.

Поступила 25.11.04

Белорусский государственный университет,
Минск, пр. Ф. Скорины, 4
e-mail: zolotorevichLA@bsu.by

L.A. Zolotorevich

FAULTS SIMULATION IN STRUCTURES VLSI ON VHDL

The problem of fault simulation of VLSI and systems of a digital electronics represented on VHDL is considered. The analysis of efficiency of the offered solutions is given.